CLAIMS

What is claimed is:

1. A method comprising:

receiving an initial code associated with a first framework, the first framework having an object hierarchy; and

converting the initial code to a converted code that combines the object hierarchy of the first framework with an object hierarchy of a second framework.

2. The method of claim 1 wherein the converting produces a class that inherits from a class of the first framework.

3. The method of claim 2 wherein the class of the first framework comprises a superclass of the first framework.

4. The method of claim 2 wherein the class of the first framework comprises a superclass named java.lang.Object.

5. The method of claim 2 wherein the class of the second framework comprises an array class.

6. The method of claim 2 wherein the class of the second framework comprises an array class named System.Array.

7. The method of claim 1 wherein the converting includes creating a new class.

8. The method of claim 7 wherein the new class inherits from java.lang.Object and from System.Array.

9. A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having an object hierarchy, to a converted code that combines the object hierarchy of the first framework with an object hierarchy of a second framework.

10. A method comprising:

receiving an initial code associated with a first framework, the first framework having an exception hierarchy; and

converting the initial code to a converted code that combines the exception hierarchy of the first framework with an exception hierarchy of a second framework.

11. The method of claim 10 wherein the converting includes mapping exceptions.

12. A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having an exception hierarchy, to a converted code that combines the exception hierarchy of the first framework with an exception hierarchy of a second framework.

13. A method comprising:

receiving an initial code associated with a first framework, the first framework having an exception hierarchy; and

converting the initial code to a converted code that maps the exception hierarchy of the first framework to an exception hierarchy of a second framework.

14. The method of claim 13 wherein the converting includes combining exception hierarchies.

15. A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having an exception hierarchy, to a converted code that maps the exception hierarchy of the first framework with an exception hierarchy of a second framework.

16. A method comprising:

receiving an initial code associated with a first framework, the first framework having reflection transparency; and

converting the initial code to a converted code that supports the reflection transparency of the first framework on a second framework.

17. The method of claim 16 wherein the converting includes checking for methods associated with the reflection transparency of the first framework.

18. The method of claim 16 wherein the converting includes rendering a stack entry transparent.

19. A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having reflection transparency, to a converted code that supports the reflection transparency of the first framework on a second framework.

20. A method comprising:

receiving an initial code associated with a first framework, the first framework having scoping; and

converting the initial code to a converted code that supports the scoping of the first framework on a second framework.

21. The method of claim 20 wherein the converting includes marking a package scope and a protected scope associated with the first framework as a public scope on the second framework.

22. The method of claim 20 wherein the converting includes marking a package scope associated with the first framework as an assembly on the second framework.

23. The method of claim 20 wherein the converting includes marking a protected scope associated with the first framework as an assembly or a family on the second framework.

24. The method of claim 20 wherein the converting includes marking, the marking selected from a member of the group consisting of marking a protected scope associated with the first framework as an assembly or a family on the second framework; marking a package scope associated with the first framework as an assembly on the second framework; marking a package scope and a protected scope associated with the first framework as a public scope on the second framework; and combinations thereof.

25. A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having scoping, to a converted code that supports the scoping of the first framework on a second framework.

26. A method comprising:

receiving an initial code associated with a first framework, the first framework having type characteristics; and

converting the initial code to a converted code that supports the type characterisitics of the first framework on a second framework.

27. The method of claim 26 wherein the converting supports type characteristics of the first framework related to casting between real and integer types on the second framework.

28. The method of claim 26 wherein the converting supports type characteristics of the first framework related to overflow and undefined types on the second framework.

29. A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having type characteristics, to a converted code that supports the type characteristics of the first framework on a second framework.

30. A method comprising:

receiving an initial code associated with a first framework, the first framework having at least one member selected from the group consisting of object hierarchies, exception hierarchies, type characteristics, reflection transparencies, and scoping; and

converting the initial code to a converted code that supports at least one of the selected members on a second framework.

31. A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having at least one member selected from the group consisting of object hierarchies, exception hierarchies, type characteristics, reflection transparencies, and scoping, to a converted code that supports at least one of the selected members of the first framework on a second framework.